

In the Claims:

Please amend Claims 1, 5, 6, 15, 19, 24, 28, 38, 42, 47, 51, 52, 61, and 65; cancel Claims 2-4, 7-14, 16-18, 20-23, 25-27, 30-37, 39-41, 43-46, 48-50, 53-60, 62-64 and 66-70; and add new Claims 71-80, all as shown below.

1. (Currently Amended) A system for providing access between an application at a mobile device, and a web service at a server, comprising:

a server comprising a processor, database, provisioning service, store and forward manager, web service, and one or more interface plugins for use by different types of mobile devices;

a mobile device comprising a memory, ~~at least one~~ processor, and software application executing thereon, wherein the mobile device further comprises

a runtime environment program, ~~executing in the~~ at least one processor, to run the application, the application containing presentation information, information for interpreting the stored application data, and information for constructing messages to a server, the runtime environment program using the information in the application to produce a display including data from the stored application data, the runtime environment program interacting with a server to update stored application data in the background when a connection between the mobile device and server is available; which displays software applications provisioned by the provisioning service, wherein the runtime environment program interacts with the server via an interface to receive the provisioned applications and asynchronously update application data between the mobile device and the database,

a software application, which is provided by the provisioning service in a markup language and which is displayable on the mobile device by the runtime environment program,

an application data store, residing in the memory of the mobile device, which is used to store

the application data for use by the software application, and templates for use by the runtime environment program in displaying the software application and the application data therein, and the application data being associated with an application, also residing in the memory,

wherein the application data being persistent at the mobile device is persisted locally with the database at the server when the connection between the mobile device and the server is available, including when the software application is either running or not running;
and

wherein the runtime environment program includes a relay interface, the relay interface knowing a transport protocol used by the mobile device and transmitting messages using the transport protocol;

wherein, while the software application executes on the mobile device, the runtime environment program sends simplified messages to the server through the relay interface, in a lower level message transport format associated with that mobile device type, to retrieve, use, or update the application data; and

wherein the server includes a transport manager-relay interface plugin, the transport manager-relay interface plugin understanding the transport protocol used by the mobile device and receiving messages transmitted to it using the transport protocol;

wherein the server receives the messages, through the transport manager-relay interface plugin configured for that mobile device type, and converts the simplified messages received from the runtime environment program into messages that are independent of mobile device type, of a first markup language based messaging protocol for the web service and converts messages of the first markup language based messaging protocol from the web service into simplified messages for the mobile device for subsequent communication to the web service, and provides responses accordingly, including using the store and forward manager to store the responses at the server until the connection between the mobile device and the server is available

wherein messaging protocol of the simplified messages is less complex than the first markup language based messaging protocol; and

wherein each simplified message consists of a title and a single block of data in a markup language format.

2-4. (Canceled).

5. (Currently Amended) The mobile device system of claim 1, wherein the mobile device and the server server use asynchronous messaging.

6. (Currently Amended) The ~~mobile device~~ system of claim 1, wherein messages are stored until a connection between the mobile device and server is available.

7-14. (Canceled).

15. (Currently Amended) The ~~mobile device~~ system of claim 1, wherein the application receives data from the web service.

16-18. (Canceled).

19. (Currently Amended) The ~~mobile device~~ system of claim 1, wherein the messages sent from the mobile device to the server are sent as a block of data in the a simplified message format which contains a fragment of a file in the markup language format.

20-23. (Canceled).

24. (Currently Amended) A method for providing access between an application at a mobile device, and a web service at a server, comprising the steps of:

providing a server comprising a processor, database, provisioning service, store and forward manager, web service, and one or more interface plugins for use by different types of mobile devices;

storing application data on a mobile device, the application data being associated with an application, the application data being persistent when the application is not running;

using a runtime environment program to execute the application on the mobile device, the application containing presentation information, information for interpreting the stored application data, and information for constructing messages to a server, the runtime environment program using the information in the application to produce a display including data from the stored application data; and

in the background, using the runtime environment program to interact with a server to update stored application data when a connection between the mobile device and server is available;

providing a mobile device comprising a memory, processor, and software application

executing thereon, wherein the mobile device further comprises

a runtime environment program, which displays software applications provisioned by the provisioning service, wherein the runtime environment program interacts with the server via an interface to receive the provisioned applications and asynchronously update application data between the mobile device and the database,

a software application, which is provided by the provisioning service in a markup language and which is displayable on the mobile device by the runtime environment program,

an application data store, residing in the memory of the mobile device, which is used to store

the application data for use by the software application, and

templates for use by the runtime environment program in displaying the software application and the application data therein, and

wherein the application data at the mobile device is persisted locally with the database at the server when the connection between the mobile device and the server is available, including when the software application is either running or not running;

wherein the runtime environment program includes a relay interface, the relay interface knowing a transport protocol used by the mobile device and transmitting messages using the transport protocol;

wherein the runtime environment program sends simplified messages to the server through the relay interface;

wherein the server includes a transport manager-relay interface plugin, the transport manager-relay interface plugin understanding the transport protocol used by the mobile device and receiving messages transmitted to it using the transport protocol;

wherein the server receives, through the transport manager-relay interface plugin, and converts the simplified messages into messages of a first markup language based messaging protocol for the web service and converts messages of the first markup language based messaging protocol from the web service into simplified messages for the mobile device;

wherein messaging protocol of the simplified messages is less complex than the first markup language based messaging protocol; and

wherein each simplified message consists of a title and a single block of data in a markup language format

sending messages, while the software application executes on the mobile device, from the runtime environment program to the server through its interface, in a lower level message transport format associated with that mobile device type, to retrieve, use, or update the application data; and receiving the messages at the server, through an interface plugin configured for that mobile device type, and converts the messages received from the runtime environment program into messages that are independent of mobile device type, for subsequent communication to the web service, and provides responses accordingly, including using the store and forward manager to store the responses at the server until the connection between the mobile device and the server is available.

25-27. (Canceled).

28. (Currently Amended) The method of claim 24, wherein the mobile device and the ~~server~~ server use asynchronous messaging.

29. (Original) The method of claim 24, wherein messages are stored until a connection between the mobile device and server is available.

30-37. (Canceled).

38. (Currently Amended) The method of claim 24, wherein the application receives data from the web service.

39-41. (Canceled).

42. (Currently Amended) The method of claim 24, wherein the messages sent from the mobile device to the server are sent as a block of data in the a simplified message format which contains a fragment of a file in the markup language format.

43-46. (Canceled).

47. (Currently Amended) A computer readable ~~media comprising a runtime environment~~

program and an application to instruct a mobile device to do the steps of medium, including instructions stored thereon which when executed cause the computer to perform the steps comprising:

providing a server comprising a processor, database, provisioning service, store and forward manager, web service, and one or more interface plugins for use by different types of mobile devices;

storing application data on a mobile device, the application data being associated with the application, the application data being persistent when the application is not running;

using a runtime environment program to execute the application on the mobile device, the application containing presentation information, information for interpreting the stored application data, and information for constructing messages to a server, the runtime environment program using the information in the application to produce a display including data from the stored application data; and

in the background, using the runtime environment program to interact with a server to update stored application data when a connection between the mobile device and server is available;

wherein the runtime environment program includes a relay interface, the relay interface knowing a transport protocol used by the mobile device and transmitting messages using the transport protocol;

wherein the runtime environment program sends simplified messages to the server through the relay interface;

wherein the server includes a transport manager-relay interface plugin, the transport manager-relay interface plugin understanding the transport protocol used by the mobile device and receiving messages transmitted to it using the transport protocol;

wherein the server receives, through the transport manager-relay interface plugin, and converts the simplified messages into messages of a first markup language based messaging protocol for the web service and converts messages of the first markup language based messaging protocol from the web service into simplified messages for the mobile device;

wherein messaging protocol of the simplified messages is less complex than the first markup language based messaging protocol; and

wherein each simplified message consists of a title and a single block of data in a markup language format

receiving messages at the server, from a mobile device comprising a memory, processor, and software application executing thereon, in a lower level message transport format associated with that mobile device type, to retrieve, use, or update the application data wherein the mobile device further comprises

a runtime environment program, which displays software applications provisioned by the provisioning service, wherein the runtime environment program interacts with the server via an interface to receive the provisioned applications and asynchronously update application data between the mobile device and the database,

a software application, which is provided by the provisioning service in a markup language and which is displayable on the mobile device by the runtime environment program,

an application data store, residing in the memory of the mobile device, which is used to store

the application data for use by the software application, and

templates for use by the runtime environment program in displaying the software application and the application data therein, and

wherein the application data at the mobile device is persisted locally with the database at the server when the connection between the mobile device and the server is available, including when the software application is either running or not running; and

converting the messages received from the runtime environment program into messages that are independent of mobile device type, for subsequent communication to the web service, and providing responses accordingly, including using the store and forward manager to store the responses at the server until the connection between the mobile device and the server is available.

48-50. (Canceled).

51. (Currently Amended) The computer readable ~~media~~ medium of claim 47, wherein the mobile device and the sever use asynchronous messaging.

52. (Currently Amended) The computer readable ~~media~~ medium of claim 47, wherein messages are stored until a connection between the mobile device and server is available.

53-60. (Canceled).

61. (Currently Amended) The computer readable ~~media~~ medium of claim 47, wherein the application receives data from the web service.

62-64. (Canceled).

65. (Currently Amended) The computer readable ~~media~~ medium of claim 47, wherein the messages sent from the mobile device to the server are sent as a block of data in ~~the a~~ simplified message format which contains a fragment of a file in the markup language format.

66-70. (Canceled).

71. (New) The system of claim 1 wherein the system supports a plurality of different types of mobile devices, by providing at the server a transport manager and a different plugin for each mobile device type, which understands the lower level message transport format or protocol used by the type of mobile device.

72. (New) The system of claim 1 wherein the the application data for use by the software application, and the templates for use by the runtime environment program in displaying the software application, are stored separately within the mobile device, so that new data can be received from the server without having to generate new markup language for displaying the software application.

73. (New) The system of claim 1, wherein the application data store includes a list of available software applications on the server.

74. (New) The system of claim 1, wherein the software application running within the runtime environment program includes a collection of screens containing the user interface information and message handlers which process messages received from the server.

75. (New) The system of claim 1, wherein the software application running within the runtime

environment program can operate as multiple independent instances of the same software application, and wherein each multiple independent instance receives a unique incarnation ID, which is used by the runtime environment program to uniquely associate data received from the server with a particular instance (and potentially user) of the software application on the mobile device.

76. (New) The method of claim 24, further comprising supporting a plurality of different types of mobile devices, by providing at the server a transport manager and a different plugin for each mobile device type, which understands the lower level message transport format or protocol used by the type of mobile device.

77. (New) The method of claim 24, wherein the the application data for use by the software application, and the templates for use by the runtime environment program in displaying the software application, are stored separately within the mobile device, so that new data can be received from the server without having to generate new markup language for displaying the software application.

78. (New) The method of claim 24, wherein the application data store includes a list of available software applications on the server.

79. (New) The method of claim 24, wherein the software application running within the runtime environment program includes a collection of screens containing the user interface information and message handlers which process messages received from the server.

80. (New) The method of claim 24, wherein the software application running within the runtime environment program can operate as multiple independent instances of the same software application, and wherein each multiple independent instance receives a unique incarnation ID, which is used by the runtime environment program to uniquely associate data received from the server with a particular instance (and potentially user) of the software application on the mobile device.